

VISOKO UČILIŠTE ALGEBRA
VISOKA ŠKOLA ZA PRIMIJENJENO RAČUNARSTVO

ZAVRŠNI RAD
KORIŠTENJE GRAFIČKIH KARTICA U
OBRADI ZVUČNOG ZAPISA

Kruno Karlović

Zagreb, veljača 2018.

Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada.

U Zagrebu 20. veljače 2018.

Predgovor

Upis na Visoku školu za primijenjeno računarstvo za mene je značio početak puta prema ostvarenju dječaćkih ambicija. Želja da nekakva vrsta rada na računalu postane moje životno zanimanje probudila se još u djetinjstvu, zajedno s prvim satima samostalno provedenim za računalom. Otvaranje smjera Multimedijskog računarstva te moje upisivanje baš toga smjera bio je još jedan korak prema ostvarenju tih mladenačkih želja.

Put ka ostvarenju tih želja nije uvijek bio jednostavan, a sada kada mi do cilja preostaje samo obrana završnoga rada, u sjećanje mi naviru svi trenuci koji su tome prethodili, i on ugodni kao i oni možda manje ugodni, ali jednako važni na putu prema ostvarenju toga cilja.

Prije svega, zahvalu za uspjeh dugujem svojoj obitelji, majci Vesni i dugogodišnjoj djevojci, a danas supruzi Sanji koje su me uvijek poticale da idem naprijed i bile bezrezervna podrška u svemu.

Zahvaljujem i prijateljima i kolegama s fakulteta koji su učinili studentske dane veselim i zanimljivima, a ujedno su bili i podrška u otežavajućim okolnostima koje su se kroz studentske dane pojavljivale.

Također bih zahvalio profesorima i asistentima na studiju koji su prenosili svoje znanje na nas studente i bili uvijek spremni pomoći i dodatno objasniti određene probleme kada je to bilo potrebno.

Na kraju bih se zahvalio profesoru i mentoru, gospodinu Vedranu Dakiću, koji me je usmjeravao i pomagao kako bih izradio ovaj završni rad.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvatanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

U završnom radu želja mi je proučiti mogućnosti obrade zvuka pomoću grafičkih kartica korištenjem CUDA dodataka (engl. *plug-in*). Cilj je rada na praktičnom primjeru stvarnog audio zapisa primijeniti *plug-inove* bazirane isključivo na CUDA tehnologiji te usporediti takav vid obrade zvuka s klasičnom obradom putem hardverskih *plug-inova*.

Ključne riječi: obrada zvuka, grafička kartica, CUDA, *plug-in*

In the final paper, my desire is to study the possibilities of sound processing with use of graphics cards by using CUDA plug-ins. The goal of this paper is to apply plug-ins based on CUDA technology on example of a real audio track, and to compare such sound processing aspect with classical processing via hardware plug-ins.

Key words: sound processing, graphics card, CUDA, plug-in

Sadržaj

1. Uvod	1
2. CUDA.....	2
2.1. CUDA arhitektura	5
2.1.1. CUDA programski model.....	5
2.1.2. CUDA memorijski model.....	6
2.1.3. CUDA model radnog toka.....	7
2.2. Primjena CUDA-e	8
2.2.1. Izrada medicinskih snimaka	9
2.2.2. Okolišna znanost	10
2.2.3. Računalna dinamika fluida	11
2.2.4. Duboko učenje.....	12
3. Primjena CUDA <i>plug-ina</i> na zvučni zapis	14
3.1. DAW (Digital Audio Workstation)	17
3.1.1. Softverski DAW	17
3.1.2. <i>Plug-inovi</i>	19
4. Rezultati.....	21
5. Zaključak	26
Popis kratica	27
Popis slika.....	28
Literatura	29
Prilog	30

1. Uvod

Odabir ove teme kao završnoga rada došao je iz želje za stjecanjem novih iskustava iz područja obrade zvuka. Samo područje obrade zvuka oduvijek me zanimalo, a taj interes je samo porastao polaskom na kolegij Elektroakustika i profesionalna audio oprema. Vrlo brzo sam odlučio da ću i završni rad raditi upravo iz toga kolegija. Nakon što mi je mentor predložio nekoliko tema, odabrana tema mi se učinila najzanimljivijom i odlučio sam samostalno malo dublje proučiti navedeno područje obrade zvuka.

S obzirom na manjak osobnoga iskustva u praktičnoj obradi zvuka, mentorovo znanje i predložena literatura bili su veliki zamah za početak izrade ovoga rada.

U uvodnim poglavljima ovoga rada nalazi se teoretsko upoznavanje s tehnologijom korištenom prilikom izrade istoga. Nakon toga slijedi opis praktičnoga testiranja navedene tehnologije, na primjeru glazbenog audiozapisa.

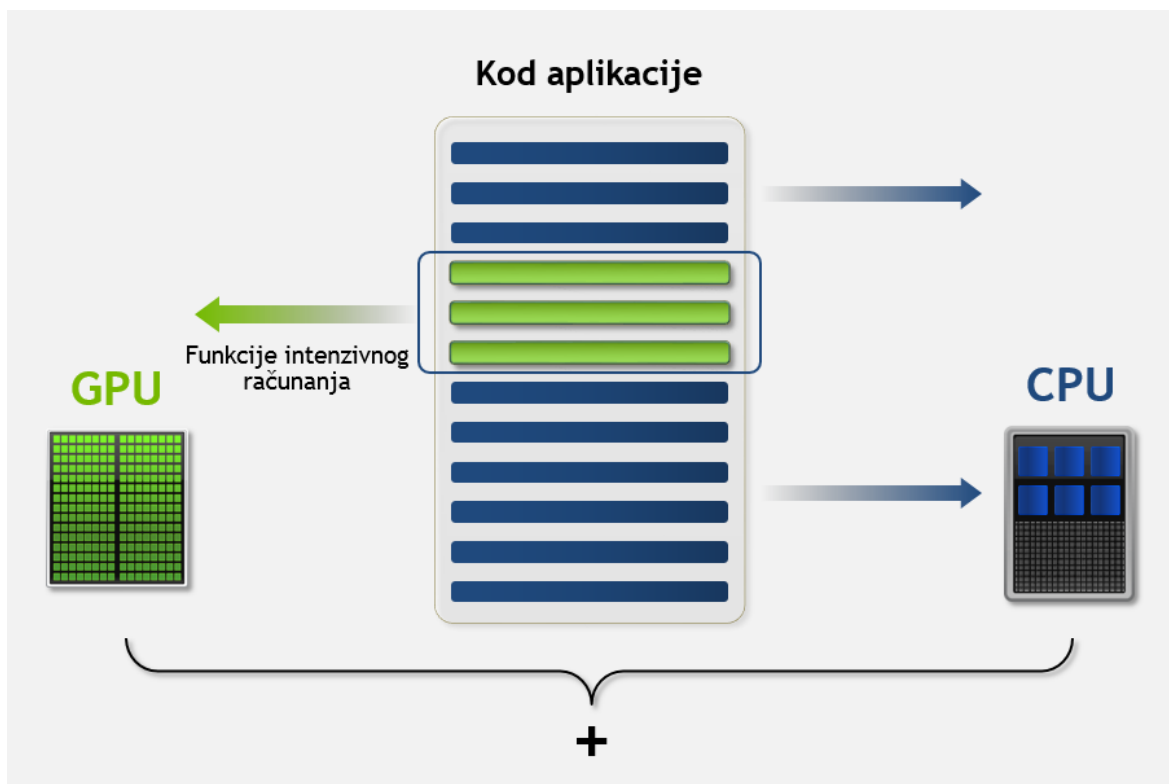
U prvom dijelu praktičnoga testiranja audiozapis je obrađivan isključivo korištenjem CUDA dodataka (engl. *plug-in*), dok u drugom dijelu testiranja isti taj audiozapis prolazi iste metode obrade, ali korištenjem hardverskih *plug-inova*.

Krajnji je cilj usporediti ova dva pristupa obradi zvuka te donijeti zaključak o eventualnim prednostima i manama korištenja CUDA *plug-inova* u odnosu na stariji, hardverski pristup.

2. CUDA

Računanje (engl. *computing*) pomoću grafičkih procesora (engl. *graphical processing unit*, skraćeno GPU) vrsta je heterogenoga računanja, odnosno predstavlja paralelno računanje s više procesorskih jedinica. Prilikom GPU računanja dolazi do povezivanja mnogojezgrenih (engl. *many-core*) grafičkih procesora s višejezgrenim (engl. *multicore*) procesorima (engl. *central processing unit*, skraćeno CPU) kako bi se postigle što bolje performanse.

Današnji CPU-i sadrže po 4, 6, 8 ili 12 jezgri (engl. *core*), dok GPU može sadržavati i do 512 jezgri u samo jednome čipu. S obzirom na to da današnje računalne aplikacije sadrže velike količine podataka koje je potrebno paralelno obraditi, GPU-ovi ubrzavaju te procese raspoređivanjem računanja na stotine jezgri što donosi značajno ubrzanje kod mnogih aplikacija. Kako bi dodali GPU ubrzanje, potrebno je prepoznati područja koja su kritična za performanse aplikacije i primijeniti GPU ubrzanje na detektirana područja. U većini slučajeva nema potrebe da se cijela aplikacija prebacuje na GPU računanje, stoga ostatak procesa nastavlja raditi preko CPU-a, kao što je prikazano na slici Slika 2.2.1. Stoga je najbitniji zaključak ovakvoga računanja to da GPU i CPU rade zajedno dok je na programeru da prepozna koje funkcije treba ubrzati korištenjem GPU-a.



Slika 2.2.1 Paralelni rad CPU-a i GPU-a.

Testiranja i mjerenja u stvarnim okolnostima pokazuju da primjena ovakvoga pristupa kod kompleksnih matematičkih i znanstvenih aplikacija, kao što je na primjer MATLAB, donosi ubrzanje od 8 do 18 puta. To bi konkretno značilo da bi za posao koji je na CPU-u trajao mjesec dana, ovim pristupom bilo dovoljno manje od tjedan dana, što je zaista velika ušteda vremena koja omogućava brži prelazak na nove zadatke unutar određenoga projekta.

CUDA (engl. *compute unified device architecture*) jest platforma za paralelno računanje koju je razvila američka tvrtka NVIDIA kako bi omogućila računanje na grafičkim jedinicama i postigla dramatično ubrzanje računalnih aplikacija korištenjem snage GPU-ova [4]. Prvi GPU s NVIDIA CUDA arhitekturom bio je GeForce 8800 GTX. Nova arhitektura sadržavala je komponente dizajnirane isključivo za GPU računanje koje su ublažile mnoga ograničenja poznata kod prijašnjih generacija GPU-ova prilikom primjene za računanje opće namjene (engl. *general-purpose computing*).

Postoje tri glavna pristupa za dodavanje GPU ubrzavanja aplikacijama:

- Ubacivanje unaprijed definiranih i optimiziranih GPU ubrzanih biblioteka (engl. *libraries*) kao alternativa MKL, IPP i ostalim često korištenim bibliotekama.

Razvoj aplikacija oko biblioteka pruža višestruke prednosti. S obzirom na to da ih je moguće pozvati iz *host* aplikacije, biblioteke ubrzane CUDA platformom imaju sposobnost "skrivanja" velikoga dijela arhitekture GPU-a i na taj način skratiti proces razvoja aplikacije. CUDA ubrzane biblioteke često koriste API-je slične onima koji se već koriste u bibliotekama korištenima za znanstvene aplikacije te stoga omogućuju promptno ubrzanje. Ponekad će biblioteke imati identično sučelje i za GPU ubranu verziju i za CPU verziju što rezultira kodom primjenjivim u oba slučaja, dok su ponekad potrebne sitne preinake u kodu za dvostruku primjenjivost. CUDA ubrzane biblioteke precizno su prilagođene za poboljšanje performansi na svim generacijama GPU-ova. Neke od poznatijih GPU ubrzanih biblioteka su:

- NVIDIA cuBLAS
 - NVIDIA cuRAND
 - NVIDIA cuSPARSE
 - NVIDIA NPP
 - CULA
 - CUSP.
- Dodavanje određenih smjernica (engl. *directives*) kompilatoru (engl. *compiler*) u izvorni kod (engl. *source code*) za automatsko paraleliziranje petlji.
OpenACC programski je standard za paralelno računanje na akceleratorima poput GPU-ova korištenjem kompilatorskih smjernica. Smjernice ukazuju kompilatoru na paralelizme u kodu kako bi izvršio mapiranje računanja na paralelne akcelerateore kao što je GPU, a mogu sadržavati i dodatne informacije. Kompilatori koji ne podržavaju OpenACC ignorirat će smjernice kao da se radi o komentarima u kodu.
 - Korištenje programskih jezika za implementaciju vlastitih paralelnih algoritama.
Programerima vještima u C++ programiranju ovakav pristup omogućuje sigurno najveću fleksibilnost u radu. Thrust je *open-source* biblioteka koja nalikuje na C++ STL biblioteku s kojom su dobro upoznati. Thrustovo sučelje olakšava produktivnost programera, a omogućuje prenosivost performansi između GPU-ova i višejezgrenih CPU-ova.

Za svaki segment aplikacije koji pokazuje slabije performanse moguće je koristiti navedene pristupe samostalno ili zajedno što znači da možemo koristiti GPU ubranu biblioteku za početne izračune te zatim napisati vlastiti kod kako bi izvršili izračune koji nam još nisu dostupni u biblioteci. CUDA platforma podržava sva tri navedena pristupa.

2.1. CUDA arhitektura

Za razliku od prethodnih generacija GPU-ova koje su dijelile računalne resurse između tjemelih (engl. *verts*) i piksel *shadera*, CUDA arhitektura uključuje jedinstveni *shader* cjevovod (engl. *pipeline*) koji omogućuje da se svaka pojedina aritmetičko logička jedinica (engl. *arithmetic logic unit*, skraćeno ALU) na čipu koristi programom koji namjerava obavljati računanje opće namjene. S obzirom na to da je NVIDIA novu generaciju GPU-ova namijenila za računanje opće namjene, korišteni ALU-ovi su izrađeni tako da zadovoljavaju uvjete IEEE-a za *single precision* aritmetiku s pomičnim zarezom (engl. *floating-point arithmetic*) te su dizajnirani za korištenje seta instrukcija prilagođenog za općenito računanje više nego za grafički prikaz. Nadalje, izvršne jedinice na GPU-u imaju mogućnost proizvoljnog pristupa za čitanje i pisanje u memoriji, kao i pristup softverski upravljanoj *cacheu* znanome kao dijeljena memorija. Sve ove značajke CUDA arhitekture dodane su kako bi kreirali GPU koji bi, osim u standardnim grafičkim zadacima, pokazivao odlične performanse i pri računanju opće namjene. Kako bi dosegli što veći broj razvojnih programera, NVIDIA je prihvatila C standard te dodala mali broj ključnih riječi (engl. *keywords*) kako bi iskoristila neke od posebnih značajki CUDA arhitekture. Nekoliko mjeseci nakon objave GeForce 8800 GTX-a, NVIDIA javno predstavlja *compiler* za ovaj jezik pod imenom CUDA C. CUDA C prvi je jezik koji je osmislila GPU tvrtke kako bi se olakšalo računanje opće namjene na GPU-ovima [5].

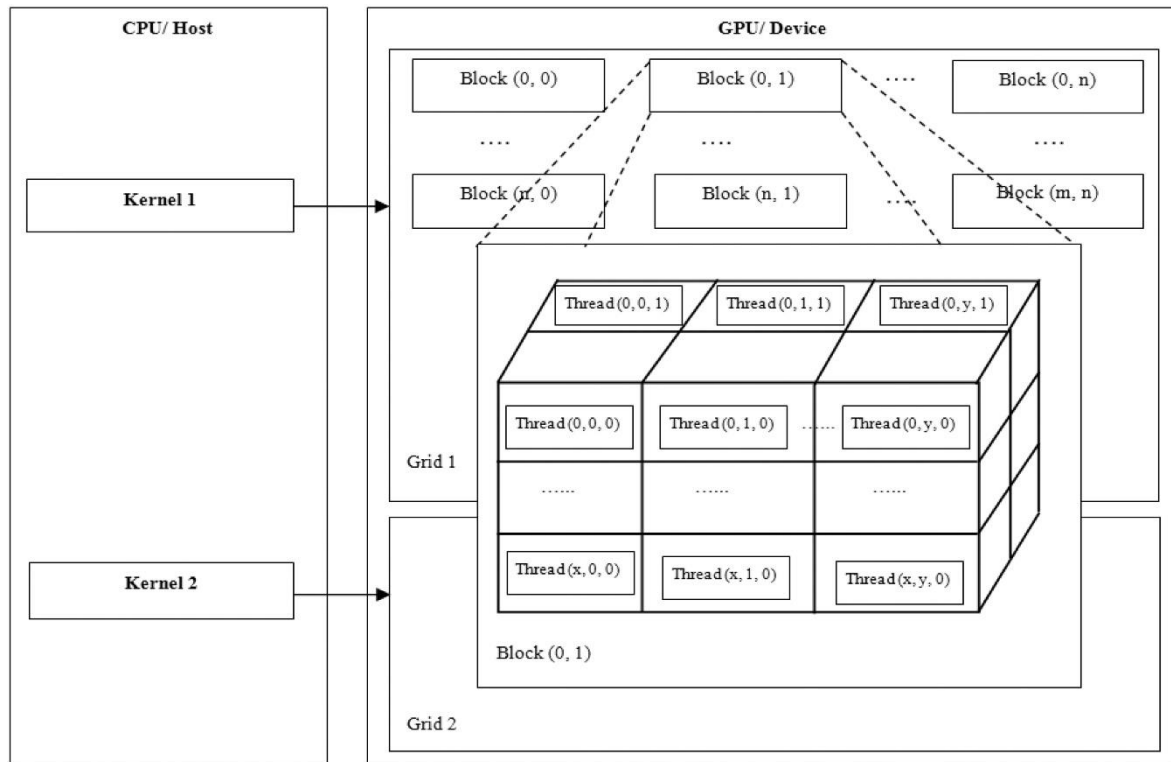
CUDA GPU ima tri tipa arhitektonske hijerarhije:

- programski model
- memorijski model
- CUDA radni tok (engl. *workflow*)

2.1.1. CUDA programski model

CUDA hardver građen je od tri glavna dijela kako bi učinkovito koristio potpunu računalnu sposobnost GPU-a. CUDA arhitekturu grade rešetke (engl. *grid*), blokovi i niti (engl. *threads*) kao što je prikazano na slici Slika 2.2. Na sve tri razine hijerarhijske arhitekture izvršenje je neovisno od ostalih entiteta na istoj razini hijerarhije. Rešetka je izrađena od skupa blokova koji se mogu samostalno izvršavati. Blokovi su organizirani kao trodimenzionalni nizovi niti te svaki blok sadrži jedinstveni ID. *Thread* se izvršava

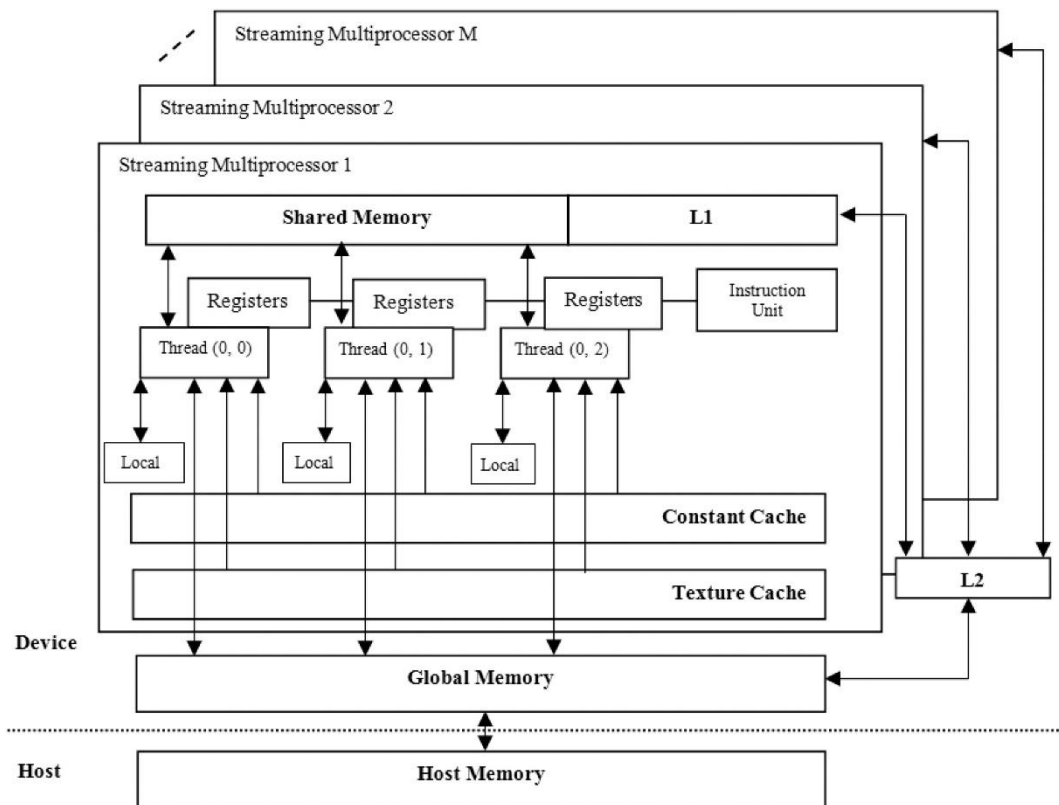
preko funkcije kernela te također sadrži jedinstveni ID. Ukupna veličina bloka ograničena je na 1024 *threada*.



Slika 2.2 Programski model CUDA GPU-a

2.1.2. CUDA memorijski model

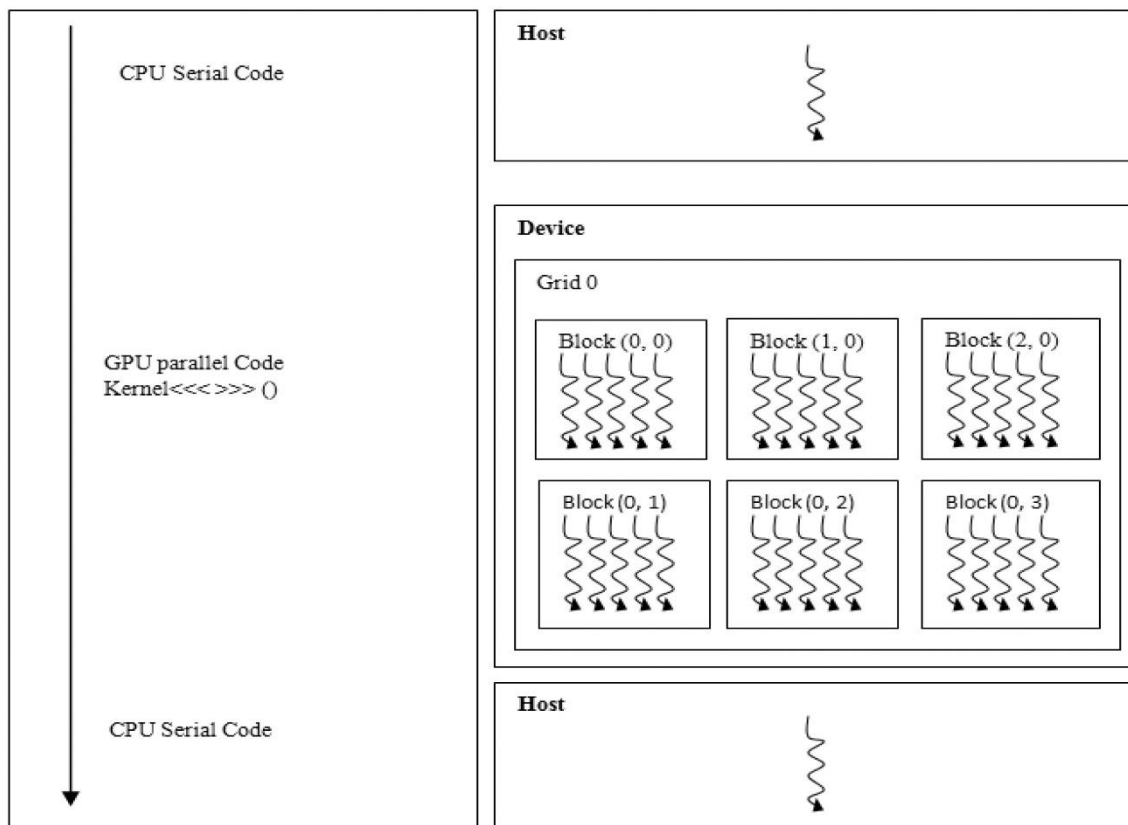
GPU koristi višestruke memorije za izvršavanje *threadova*. Na slici Slika 2.3 prikazana je organizacija CUDA memorije. GPU ima M broj tekućih multiprocesora (engl. *streaming multiprocessor*, skraćeno SM) i N broj tekućih procesorskih jezgri (engl. *streaming processor cores*, skraćeno SPs) po svakome SM-u. Svaki *thread* može pristupiti varijablama iz lokalne memorije i registara koji imaju najveću propusnost (engl. *bandwidth*) i u kojima su uglavnom spremljene često korištene varijable. Svaki pojedini blok ima vlastitu dijeljenu memoriju od 16 ili 48 kB kojoj može pristupiti svaki *thread* unutar toga bloka. Stalna (engl. *constant*) memorija podržava *read-only* pristup sa kratkim kašnjenjem i visokom propusnošću kada svi *threadovi* istovremeno pristupe istoj lokaciji. Stalni *cache* je ograničen na 64 kB. Teksturna (engl. *texture*) memorija se koristi u vizualizacijskom procesu, a ponekad može biti korištena kao oblik *cachea* kako bi se izbjegla ograničenja propusnosti globalne memorije. Globalna memorija je najveća te podržava operacije čitanja i pisanja sa niskom propusnošću. Globalna i konstantna memorija podržavaju *host* kod koji može slati podatke prema i od uređaja.



Slika 2.3 Memorijski model CUDA GPU-a

2.1.3. CUDA model radnog toka

Threadovi GPU-a imaju manju težinu od *threadova* CPU-a. CUDA program počinje sa izvršenjem domaćina (engl. *host*) dok kernel funkcija generira veliku količinu *threadova* kako bi izvršila podatkovni paralelizam. Prije pokretanja kernela, svi su bitni podaci prebačeni sa domaćina na memoriju uređaja. CPU pokreće kernel funkciju i protokol izvršenja je prebačen na uređaj. Preneseni podaci vraćaju se domaćinu na daljnju obradu. Tok CUDA izvršenja prikazan je na slici Slika 2.4.



Slika 2.4 CUDA radni tok

2.2. Primjena CUDA-e

Od samih početaka u 2007. godini, razne grane industrije sa širokim rasponom aplikacija postigle su velike uspjehe odlukom osmišljavanja aplikacija u CUDA C-u, postižući velika poboljšanja u odnosu na dotadašnje najsuvremenije implementacije. Neke od uspješnih primjena CUDA C-a i CUDA arhitekture su [6]:

- medicinska istraživanja
- naftna istraživanja
- računalna dinamika fluida (engl. *Computational Fluid Dynamics*, skraćeno CFD)
- okolišna znanost (engl. *environmental science*)
- molekularna dinamika
- elektromagnetika i elektrodinamika
- grafički *plug-inovi*
- duboko učenje (engl. *deep learning*).

2.2.1. Izrada medicinskih snimaka

Područje medicinskih snimaka obuhvaća tehnike i procese izrade vizualnih prezentacija unutrašnjosti ljudskoga tijela zbog kliničkih istraživanja i medicinskih intervencija. Jedno je od prvih znanstvenih područja u kojemu se primjenjuju prednosti GPU računanja kako bi se ubrzalo donošenje rezultata. Kao medicinska disciplina dio je radiologije i obuhvaća različita mjerenja i tehnike, kao što su ultrazvuk, magnetska rezonanca, radiografija, endoskopija, elektrokardiografija i ostale. Broj ljudi pogođen karcinomom dojke dramatično je porastao u zadnjih dvadesetak godina. Zahvaljujući neumornim naporima mnogih liječnika, velikom broju istraživanja i općenitom porastu svijesti, prevencija i liječenje ove bolesti u istom je vremenskom razdoblju također uvelike napredovala. Za liječenje je ključna što ranija dijagnoza bolesti, njeno prepoznavanje u što ranijoj fazi.

Mamografija je jedna od najboljih tehnika za rano otkrivanje karcinoma dojke, ali ima nekoliko ograničenja. Potrebna je izrada dvije ili više snimaka koje mora razviti i očitati iskusni terapeut kako bi se tumorske stanice prepoznale. Osim toga, proces izrade rendgenskih snimaka nosi sa sobom rizike od opetovanoga zračenja pacijentova prsnoga koša. Liječnici su vrlo često prisiljeni tražiti višestruko snimanje i izradu biopsije kako bi eliminirali mogućnost postojanja karcinoma.

Ultrazvučno snimanje sigurnije je od snimanja rendgenskim zrakama i liječnici ga često koriste u kombinaciji s mamografijom kako bi bili što točniji u dijagnozi i skrbi karcinoma, ali i konvencionalno ultrazvučno snimanje pokazuje određena ograničenja. Kao rezultat svega, nastao je medicinski sustav TechniScan. TechniScan je razvio obećavajuću metodu trodimenzionalnoga ultrazvučnog snimanja, ali njihova rješenja nisu provedena u praksi zbog veoma jednostavnoga razloga, ograničenja u računanju. Najjednostavnije rečeno, pretvaranje prikupljenih ultrazvučnih podataka u trodimenzionalnu sliku zahtjeva izuzetno dugo vrijeme i preskupo je za praktičnu, svakodnevnu upotrebu.

Predstavljanje prvog GPU-a na temelju CUDA arhitekture uz njegov CUDA C programski jezik omogućilo je platformu na kojoj TechniScan može pretvoriti svoje zamisli u stvarnost. Njihov sustav za ultrazvučno snimanje Svira koristi ultrazvučne valove kako bi snimio pacijentova prsa. TechiScan Svira sustav oslanja se na dva NVIDIA Tesla C1060 procesora kako bi obradio 35 GB podataka nastalih 15-minutnim snimanjem. Zahvaljujući računalnoj snazi Tesla C1060 procesora, liječnik u roku od 20 minuta može manipulirati s izuzetno detaljnom, trodimenzionalnom snimkom dojke. Široka primjena TechniScanovog

Svara sustava počela je 2010. godine. Primjena CUDA-e za izradu medicinskih snimaka otada je pronašla veoma široku primjenu, a neki od poznatih primjera su:

- digitalna tomografija
- fluoroskopska CT rekonstrukcija
- automatski CT ultrazvuk za dijagnostičko snimanje i intervencije
- napredne MRI rekonstrukcije

2.2.2. Okolišna znanost

Povećana potreba za širokom potrošnjom ekoloških proizvoda nastala je kao prirodna posljedica brzorastuće industrijalizacije globalne ekonomije. Sve veća zabrinutost zbog klimatskih promjena, konstantan porast cijene goriva i stalni rast koncentracije onečišćujućih tvari u zraku i vodi doveli su do potrebe za reakcijom i smanjenjem štete nastale naprecima u industrijskoj proizvodnji. Deterdženti i sredstva za čišćenje već su dugo jedan od najpotrebnijih, a ujedno i ekološki najkatastrofalnijih proizvoda u svakodnevnoj uporabi. Kao rezultat, mnogi su znanstvenici počeli raditi na tome kako bi smanjili negativan utjecaj deterdženata na okoliš bez utjecanja na njihovu učinkovitost.

Ključna komponenta sredstava za čišćenje jesu surfaktanti. Surfaktantske molekule određuju kapacitet čišćenja i teksturu deterdženata i šampona, ali su često prepoznati kao za okoliš najštetnija komponenta sredstava za čišćenje. Molekule se vežu za prljavštinu i u vodi se ispiru zajedno sa česticama nečistoće. Mjerenje sposobnosti čišćenja novoga surfaktanta traži opsežna laboratorijska istraživanja na raznim kombinacijama materijala i nečistoća koje treba očistiti. Očekivano, ovakav proces iziskuje jako puno vremena i novca.

Sveučilište Temple u Philadelphiji u suradnji s tvrtkom Procter & Gamble razvilo je molekularnu simulaciju interakcije surfaktanata s vodom, prljavštinom i ostalim materijalima. Uvođenje računalnih simulacija dovodi ne samo do ubrzavanja tradicionalnoga laboratorijskog pristupa nego i proširuje širinu ispitivanja na brojne varijacije okolišnih uvjeta, daleko više nego se to moglo praktično ispitati u prošlosti. Istraživači na sveučilištu Temple koriste HOOMD simulacijski softver koji je osmislio USDOE-in Ames laboratorij. Dijeljenjem simulacije između dva NVIDIA Tesla GPU-a

postigli su performanse jednake onima na 128 CPU jezgri Cray XT3 ili 1024 CPU jezgre IBM-ova BlueGene/L stroja. Povećanjem broja Tesla GPU-ova u svom istraživanju, postigli su 16 puta brže interakcije surfaktanata nego na prethodno korištenim platformama. S obzirom na to da je NVIDIA CUDA smanjila vrijeme izvršavanja takvih kompleksnih simulacija od nekoliko tjedana na samo nekoliko sati, u narednim godinama očekuje se veliki porast u proizvodnji proizvoda s pojačanom učinkovitošću i smanjenim negativnim utjecajem na okoliš.

2.2.3. Računalna dinamika fluida

CFD je grana mehanike fluida koja analizira kretanje fluida i hidrodinamiku korištenjem numeričkih analiza i kompleksnih struktura podataka. Sam dizajn visoko efikasnih rotora i oštrica dugo je vremena bio u stagnaciji. Izuzetno kompleksno kretanje zraka i fluida oko ovih uređaja nije moguće prikazati jednostavnim formulacijama, stoga je dobivanje preciznih simulacija zbog računalne ograničenosti uglavnom ostajalo u sferi nerealnog. Samo su najveća superračunala mogla ponuditi računalne resurse zadovoljavajuće za sofisticirane numeričke modele potrebne pri razvoju i testiranju modela. S obzirom na to da su samo rijetki u mogućnosti imati pristup takvim superračunalima, inovacije u dizajnu bile su u stalnoj stagnaciji.

Dr. Graham Pullan i dr. Tobias Brandavik sa Sveučilišta u Cambridgeu prvi su prepoznali potencijal CUDA arhitekture za ubrzavanje računalne dinamike fluida na do tada nezamislive razine. Njihova početna istraživanja pokazuju da bi se prihvatljive razine performansi mogle dobiti na osobnim radnim stanicama s GPU pogonom. Kasnije se pokazalo da uporaba malih GPU klastera (engl. *cluster*) s lakoćom pokazuje bolje rezultate od skupljih superračunala. Veliki pomaci u performansama korištenjem CUDA arhitekture i dostupnost velikog broja GPU-ova doveli su do znatnog ubrzanja u istraživanjima, a mogućnost dobivanja eksperimentalnih rezultata u samo nekoliko sekundi ubrzala je postupak dobivanja povratnih informacija na koje se istraživači oslanjaju kako bi došli do otkrića. Korištenje GPU klastera potpuno je promijenilo pristup u istraživanjima, a gotovo interaktivne simulacije omogućile su velik broj novih inovacija u grani koja je dugo stagnirala.

2.2.4. Duboko učenje

Duboko učenje grana je strojnog učenja koje je posebno prikladno za rješavanje problema iz područja umjetne inteligencije. Duboko učenje temelji se na predstavljanju podataka složenih reprezentacija do kojih dolazi slijedom naučenih nelinearnih transformacija. Metode dubokog učenja svoje primjene pronalaze u izazovnim područjima gdje je dimenzionalnost podataka izuzetno velika: računalni vid, obrada prirodnog jezika ili razumijevanje govora¹. Iako strojno učenje postoji već desetljećima, dva su relativno nova trenda potaknula široku primjenu strojnog učenja: dostupnost velikih količina testnih podataka te snažno i učinkovito paralelno računanje koje pruža GPU računarstvo. Tradicionalno strojno učenje koristi algoritme strojnog učenja specifične za rukovanje rukopisom i posebne modalitete za označavanje slika ili prepoznavanje glasova. Takva metoda pokazala je nekoliko nedostataka i u vremenu potrebnom za dobivanje rješenja i u preciznosti. Današnje napredne duboke neuronske mreže koriste algoritme, velike podatke (engl. *big data*) i računalnu snagu GPU-a. GPU-ovi se koriste za osposobljavanje tih dubokih neuronskih mreža pomoću daleko većih setova treninga, u manjem vremenskom roku, koristeći daleko manje podatkovne infrastrukture. GPU-ovi se također koriste za izvođenje obučanih modela učenja strojeva za klasifikaciju i predviđanje u oblaku (engl. *cloud*), podržavajući puno veći volumen i propusnost podataka uz manje energije i infrastrukture. Neki od prvih korisnika GPU akceleratora za strojno učenje jesu velike *web* kompanije i kompanije društvenih mreža, kao i vodeće institucije u znanosti o podacima (engl. *data science*) i strojnom učenju. S tisućama računalnih jezgri i protokom od deset do sto puta većim nego kod samo CPU-a, GPU-i su postali procesi koje se koristi za obradu velikih količina podataka².

Algoritmi dubokog učenja koriste velike količine podataka i računalnu snagu GPU-a kako bi saznali informacije izravno iz podataka poput slika, signala i teksta. NVIDIA DIGITS nudi interaktivno rješenje za kompleksne probleme klasifikacije slika. Okvir (engl. *framework*) za duboko učenje nudi veću fleksibilnost pri izradi i treningu prilagođenih dubokih neuronskih mreža i pruža sučelje za uobičajene programske jezike. DIGITS omogućava brzo dizajniranje najbolje duboke neuronske mreže (engl. *deep neural network*,

¹ <https://developer.nvidia.com/deep-learning>

² <http://www.nvidia.com/object/machine-learning.html>

skraćeno DNN) za podatke, interaktivno i bez pisanja koda, kako bi se postigli najsvremeniji rezultati s dubokim učenjem³.

NVIDIA Deep Learning SDK pruža moćne alate i biblioteke za projektiranje i implementaciju GPU ubrzanih aplikacija za duboko učenje. To uključuje biblioteke za zaključivanje (cuDNN), video analitiku (DeepStream SDK), linearnu algebru (cuBLAS), rijetke matrice (cuSPARSE) i multi-GPU komunikaciju (NCCL). Deep Learning SDK zahtijeva CUDA set alata (engl. *toolkit*), koji nudi sveobuhvatno razvojno okruženje za izgradnju novih GPU ubrzanih algoritama dubokog učenja i dramatično povećanje performansi postojećih aplikacija. NVIDIA Deep Learning SDK ubrzava raširene *frameworke* za duboko učenje kao što su Caffe, CNTK, TensorFlow, Theano i Torch, kao i mnoge druge aplikacije za duboko učenje⁴.

³ <https://developer.nvidia.com/digits>

⁴ <https://developer.nvidia.com/deep-learning-software>

3. Primjena CUDA *plug-ina* na zvučni zapis

Tema i cilj ovoga rada bili su istražiti primjenu CUDA-e na obradu zvučnoga zapisa korištenjem CUDA audio *plug-ina*. Konverzijski reverbi su primarni kandidati za korištenje GPU-a, budući da im je potrebna velika količina sličnih kalkulacija u kratkom roku, stoga ne čudi da je jedan od prvih audioproizvoda dostupan u CUDA izdanju bio upravo konverzijski reverb koji je osmislila tvrtka LiquidSonics. Njihov Reverberate LE dostupan je u prirodnoj (engl. *native*) i CUDA verziji, iako njihovi programeri upozoravaju da ovisno o tome koju NVIDIA grafičku karticu koristite, *native* verzija može biti učinkovitija, pogotovo ako se koristi manji audio pufer za manju latenciju. Razlog za to jest dodatna obrada blokova podataka prema GPU-u i od GPU-a. Nebula3 Pro tvrtke Acustica Audio nudi potpunu snagu njihove tehnologije obrade zvuka, a moguće ga je izvoditi pomoću CUDA tehnologije, oslobađajući CPU od napora⁵.

Nebula3 Pro je *plug-in* s mnoštvom efekata koji može emulirati i replicirati razne vrste skupe audioopreme, eliminirajući na taj način potrebu za skupim i glomaznim hardverom. Nebula koristi biblioteke kreirane sofisticiranim pristupom sempliranja koji omogućava snimanje aspekata zvuka audiouređaja i njihovu reprodukciju. Rezultat ovakvog procesa sempliranja veoma je realistična i kompleksna emulacija raznih obrada zvučnoga signala, snimljena i korištena unutar *plug-ina* koji je kompatibilan s većinom DAW-ova i softvera za obradu. Nebula sadrži preko 400 realnih hardverskih emulacija uključujući pretpojačala, ekvalizatore (engl. *equalizer*), kompresore, trake, filtere, reverbe i 90 varijacija efekata kao što su zbor, *flanger* i *phaser*⁶.

Prilikom izrade ovoga rada koristio sam Nebula3 Free, besplatnu verziju Nebula3 Pro-a, prikazanu na slici Slika 3.3.1 Slika 3.3.1 Nebula3 Free.

⁵ <https://www.soundonsound.com/sound-advice/using-your-graphics-card-process-plug-ins>

⁶ <https://www.kvraudio.com/product/nebula3-pro-bundle-by-acustica-audio>



Slika 3.3.1 Nebula3 Free

Osim Nebula3 Free *plug-ina*, koristio sam se i sljedećim:

- Nebula3 Free Reverb
- AcquaVox
- Red Eq
- Stradipad Free.

AcquaVox je *plug-in* jednostavan za korištenje, ali je sjajna emulacija klasičnog, visoko cijenjenog pretpojačala. Odličan je za dodavanje topline snimljenim instrumentima, poput gitara i bubnjeva te za prenošenje analognoga osjećaja na vokalnim snimkama. Prikazan je na slici Slika 3.3.2. AcquaVox je besplatan proizvod i jedan je od prvih samostalnih *plug-ina* iz Nebula linije⁷.



Slika 3.3.2 AcquaVox

⁷ <https://www.kvraudio.com/product/acquavox-by-acustica-audio>

Red Eq je besplatni *equalizer* koji pruža tri fiksne frekvencije od 80 Hz, 1.8 kHz i 8 kHz, prikazan na slici Slika 3.3.3. Nudi opseg od 16 dB za smanjenje i pojačavanje, i iako EQ možda djeluje limitirano u odnosu na parametarski, zvuči glazbeno ugodno i predefinirane frekvencije su pažljivo odabrane.



Slika 3.3.3 Red Eq

Mjerač s pet razina i gumb za izlaznu dobit (engl. *gain*) omogućuju kontrolu ukupne razine izlaza s jedinice do najviše +6 dB. Crvena lampica za upozorenje pali se ako se signal približi pucanju zbog pretjeranoga korištenja već vrućeg signala. Red Eq se može koristiti kako bi se radile preinake u najpotrebnijim dijelovima zvučnoga spektra, a da zvuk ne izgubi fokus⁸.

Stradipad Free je besplatno virtualno pretpojačalo namijenjeno za obradu akustičnih žičanih instrumenata. *Plug-in* se temelji na jednoj postavci (engl. *preset*) komercijalnog Stradipad Swedish String virtualnog pretpojačala tvrtke Acustica Audio. Može se koristiti za obradu žičanih zvukova i njihovu pripremu za *mix*. Vizualni izgled *plug-ina* je prikazan na slici Slika 3.3.4. Poput svih ostalih Acustica Audio proizvoda, i Stradipad Free temelji se na njihovoj Nebula tehnologiji⁹.



Slika 3.3.4 Stradipad Free

⁸ <https://www.acustica-audio.com/store/products/red-free>

⁹ <https://bedroomproducersblog.com/2013/09/27/acustica-audio-stradipad-free/>

3.1. DAW (Digital Audio Workstation)

Digitalna audio radna stanica (engl. *Digital Audio Workstation*, skraćeno DAW) predstavlja elektronički uređaj ili softver koji se koristi za snimanje, pohranjivanje i obrađivanje audiozapisa poput glazbenih komada, pjesama, govora ili zvučnih efekata. Iako DAW dolazi u jako širokom rasponu složenosti konfiguracija koji seže od jednostavnoga softvera na kućnom računalu sve do vrlo složenih konfiguracija kontroliranih središnjim računalom, svi moderni DAW-ovi, bez obzira na konfiguraciju, imaju središnje sučelje koje korisniku omogućuje obradu i miješanje (engl. *mix*) višekanalnih audiosnimaka u konačni proizvod [1].

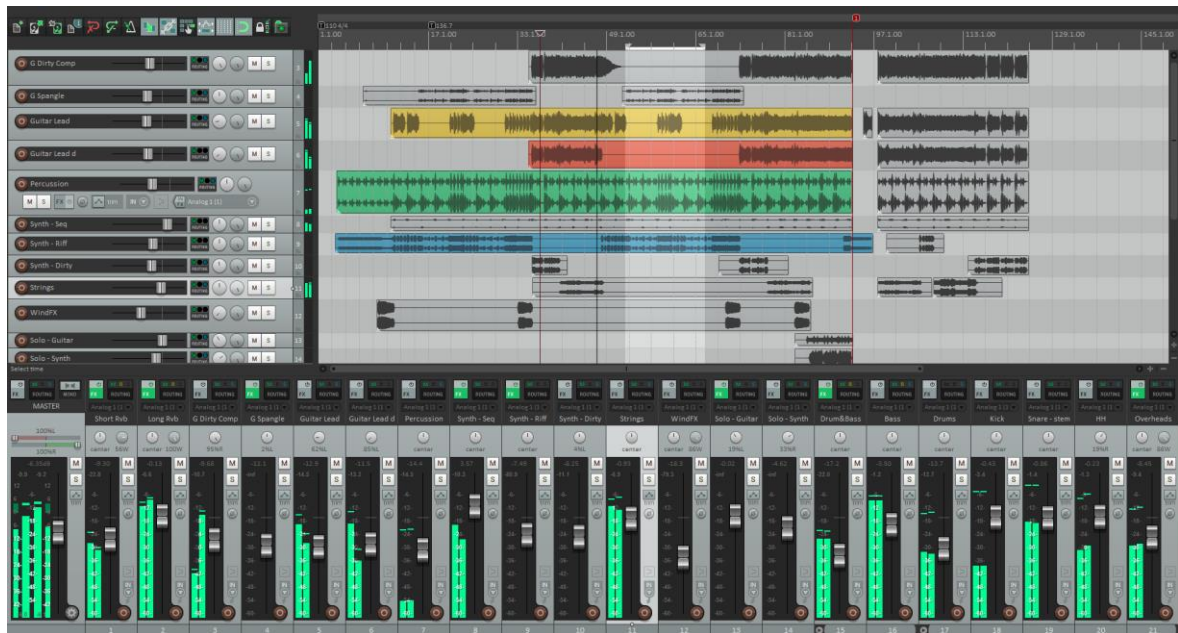
Sami počeci izrade digitalnih audio radnih stanica sežu u 1970-e, a među prvim problemima koji su se pojavili bili su znatno sporija brzina diskova te visoka cijena potrebnih komponenata i sustava za pohranu. Godine 1978. Soundstream je pod imenom Digital Editing System kreirao uređaj koji se može smatrati prvom digitalnom audio radnom stanicom koja je sadržavala analogni i digitalni audioulaz/izlaz za povezivanje sa Soundstreamovim digitalnim snimačima jednako kvalitetno kao i s analognim snimačima na vrpcu. Tijekom 80-ih godina prošloga stoljeća dolazi do velike ekspanzije izrade DAW sustava, a već početkom 90-ih velika većina glazbenih studija prelazi na digitalnu obradu zapisa što potiče još veći razvoj softverskih sustava za obradu zvuka. Integrirani DAW sustavi sastojali su se od konzole za *mix*, audiopretvarača (engl. *converter*) i sustava za pohranu podataka [2]. Zauzimali su velik prostor i cijenom su bili nedostupni većini populacije. Razvojem računala, porastom njihove moći i brzine obrade zadataka interes za skupim, integriranim DAW sustavima sve je više opadao.

3.1.1. Softverski DAW

Digitalna audio radio stanica može predstavljati samo softver za obradu zvuka, ali tehnički gledano DAW predstavlja sustav snimanja koji sadrži nekoliko komponenti:

- računalo
- softver za digitalnu obradu audiozapisa
- digitalno audiosučelje
- *plug-inove*
- DSP ubrzanje.

Računalo je sa svojim tvrdim diskovima srce DAW sustava, dok softver omogućava uređivanje zvučnoga zapisa, daje nam sučelje u kojem se sve odvija i stvara virtualni studio za obradu zvuka unutar samog računala. Zvučno sučelje pruža nam pretvaranje digitalnih signala u analogni oblik i obratno te pridonosi u daljnjoj obradi audiozapisa, a poneki DAW sustavi sadrže i dodatna audiosučelja što omogućava sve veći broj audioulaza. *Plug-inovi* su mali softverski sustavi koji rade unutar glavnoga sustava za audioobradu i proširuju njegove funkcionalnosti.



Slika 3.5 Primjer sučelja softverskog DAW-a

Danas je korisnicima dostupan zaista jako veliki izbor DAW softvera, a neki od najpoznatijih su:

- Cubase
- FL Studio
- Logic Pro
- Mixcraft
- Nuendo
- Pro Tools
- REAPER
- Studio One
- Ardour
- Mixbus.

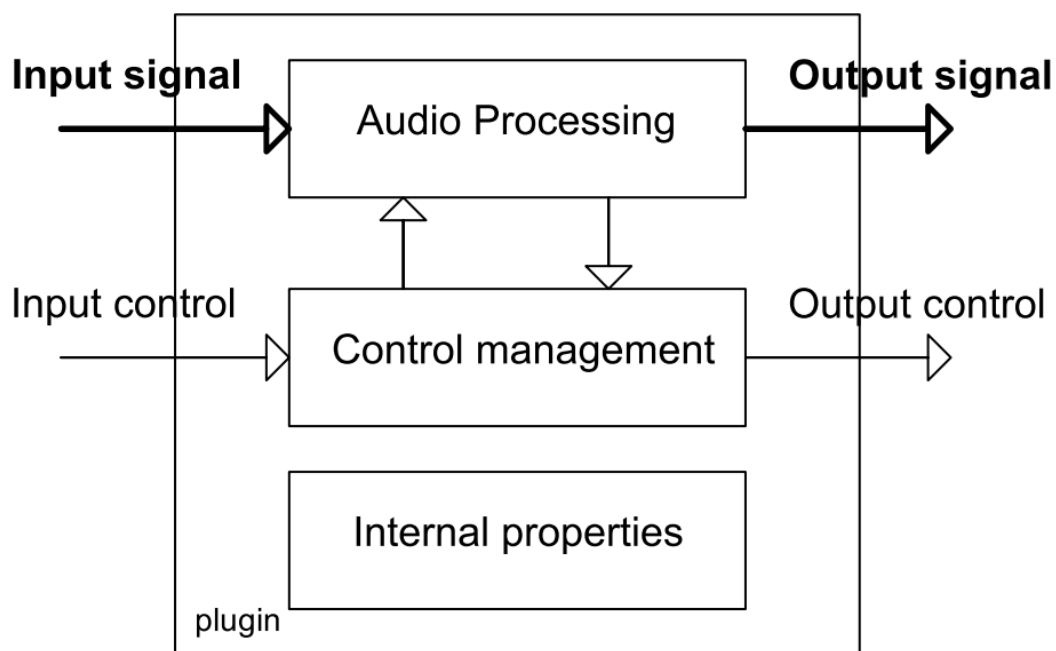
Prilikom rada na praktičnome dijelu ovoga završnoga rada upotrebljavao sam REAPER DAW.

DAW sustavi bazirani na računalu pružaju zaista skoro pa neograničene mogućnosti pri obradi zvuka, konstantno dodavanje dodatnih zapisa te virtualne klavijature i ostale instrumente prilikom snimanja glazbenih zapisa. Većina takvih sustava ima sučelje koje podsjeća na nekadašnje višetrake (engl. *multitrack*) snimače i samim time su bliski i lakše prihvatljivi glazbenicima i audioinženjerima. Primjer sučelja softverskog DAW-a prikazan je slikom Slika 3.5 Digitalne audio radne stanice pružaju nam gotovo sve funkcionalnosti kao i nekadašnji glazbeni studiji i posljednjih godina ih sve više zamjenjuju u glazbenoj industriji. Osim toga, DAW sustavi pružaju korisnicima jednu veoma bitnu funkcionalnost koju nekadašnji *multitrack* snimači nisu mogli, a to je mogućnost poništavanja posljednje radnje, sprječavajući tako neželjena trajna brisanja prethodno snimljenoga materijala ili neželjeno presnimavanje preko već postojećega zapisa.

3.1.2. *Plug-inovi*

Primarna je namjena *plug-inova* za obradu zvuka povećanje i poboljšanje mogućnosti manipulacije zvučnim zapisom u postojećem sustavu za obradu zvuka. Možemo ih smatrati virtualnim uređajima koji zamjenjuju fizičke uređaje kao što su odjek (engl. *reverb*), pojačala, kompresori, odmaci (engl. *delay*), klavijature (engl. *synthesizer*) i slično.

Plug-inove možemo podijeliti na *plug-inove* integrirane u multimedijско sučelje za programiranje aplikacija (engl. *application programming interface*, skraćeno API) i *plug-inove* povezane isključivo s matičnim softverskim platformama kao npr. RTAS (Real-Time AudioSuite). Grafički prikaz arhitekture audio *plug-inova* možemo vidjeti na slici Pogreška! Izvor reference nije pronađen.



Slika 3.6 Model arhitekture audio plug-ina

Grafičko korisničko sučelje (engl. *graphical user interface*, skraćeno GUI) *plug-ina* je sigurno podjednako važno kao i sama funkcionalnost i često je upravo sučelje razlog zašto je određeni *plug-in* popularniji ili korišteniji od konkurencije. Stoga je ključno da je korištenje jednostavno, brzo i precizno, a ujedno i vizualno jasno i dostupno korisniku. Bazično bi mogli podijeliti audio *plug-inove* na one koji vrše sintezu novoga zvučnoga zapisa, *plug-inove* za pretvorbu već postojećega zvučnoga zapisa i *plug-inove* koji analiziraju zvučni zapis. Osim *plug-inova* koji su specijalizirani isključivo za stvaranje određenih audioefekata, postoje i domaćinski (engl. *host*) *plug-inovi* koji osim sposobnosti odrađivanja specifičnoga efekta mogu biti *host* dodatnog *plug-ina* te na taj način stvarati višestruke efekte istovremeno bez zauzimanja dodatnih audiokanala na DAW sustavu [3].

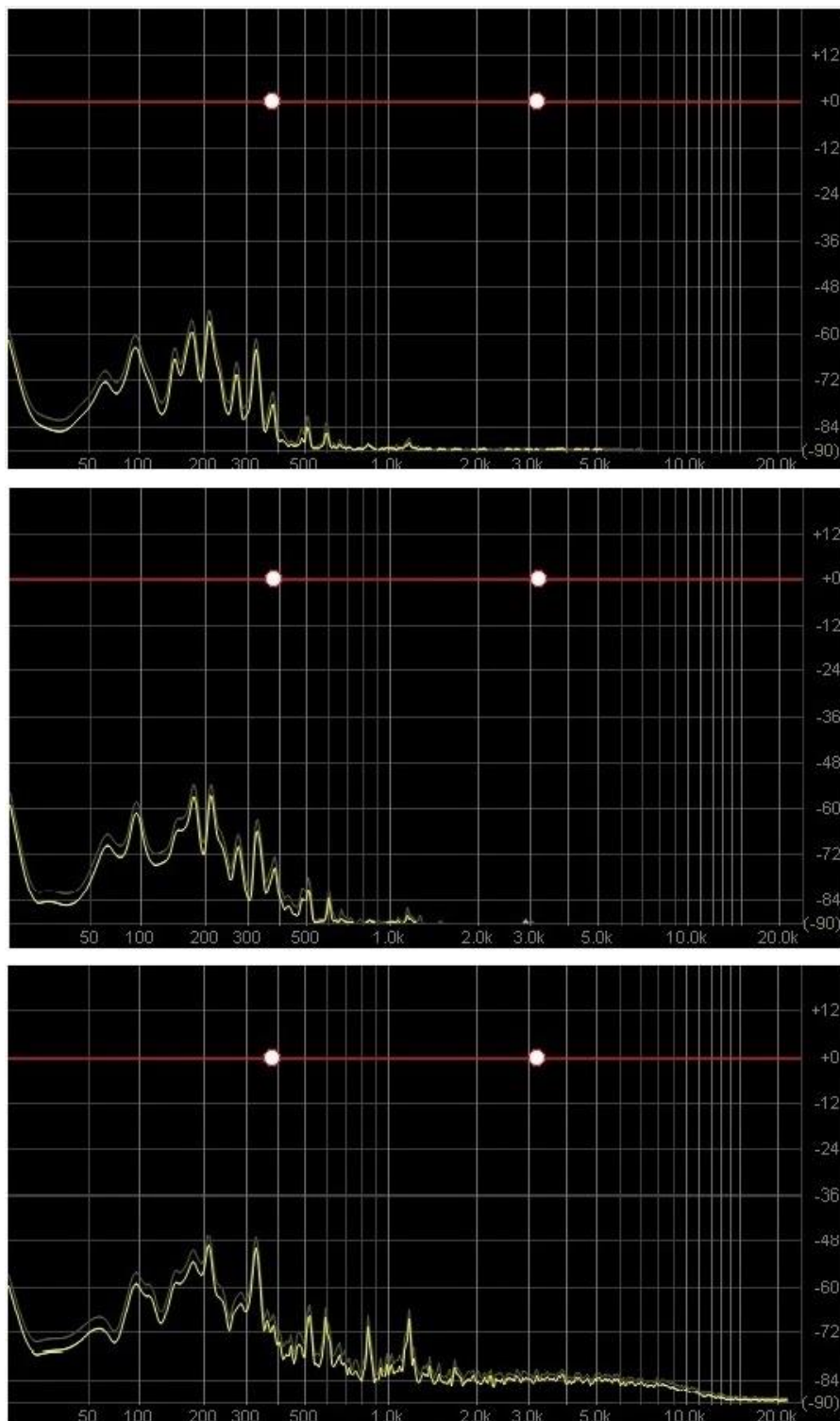
4. Rezultati

Testiranje korištenja zvučnih plug-ina baziranih na CUDA tehnologiji napravljen je na studijskoj višekanalnoj snimci, kako bi rezultate bilo što jednostavnije objasniti. Korištena je poznata pjesma od američkog kantautora Stevie Wonder-a, *Superstition*. Sama je pjesma podijeljena u šesnaest audio staza:

1. Kick (popularno bas-bubanj)
2. Overhead hi-hat (široka zvučna slika iznad *hi-hat* činele)
3. Overhead ride (široka zvučna slika iznad *ride* činele)
4. Vox1 (prva traka vokala)
5. Vox2 (druga traka vokala)
6. Clav1 (prvi Clavinet)
7. Clav2 (drugi Clavinet)
8. Clav3 (treći Clavinet)
9. Clav4 (četvrti Clavinet)
10. Clav5 (peti Clavinet)
11. Clav6 (šesti Clavinet)
12. Clav7 mic (sedmi Clavinet sniman mikrofonom)
13. Clav8 (osmi Clavinet)
14. Horn1 (trubačka sekcija, lijevo)
15. Horn2 (trubačka sekcija, desno)
16. Synth_Bass (sintetički bas)

Iako se u današnjoj produkciji obično koristi više kanala – primjera radi, snimke bubnja nerijetko zauzimaju po 30+ zvučnih staza – ovakva je zvučna slika zapravo realna s obzirom na doba kada je pjesma snimljena (1972.). U to doba nije postojala mogućnost snimanja beskonačne količine kanala i pokušaja svakog pojedinog instrumenta, i bubanj se kao instrument redovito snimao na svega 2-3 kanala (ta se tehnika i danas koristi, ali ne

više toliko često osim kod manjih sastava, tipično u jazz glazbi). Pogledajmo sada izdvojene detalje sa opisima i usporedbama.



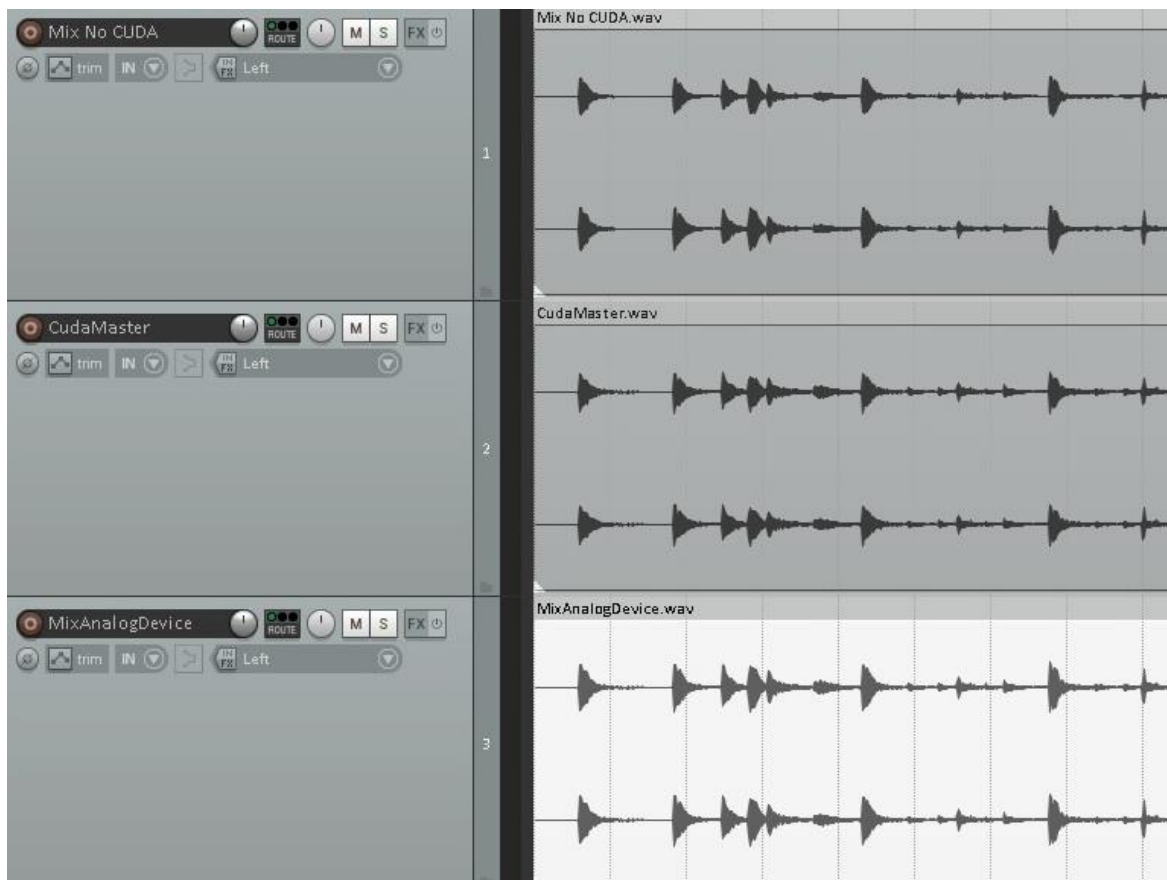
Slika 4.1. Usporedba valnog signala tri mikšana zvuka – originalnog bez efekata, miksa baziranog na CUDA softverskom plugin-u i miksa baziranog na hardverskom uređaju kojeg CUDA softverski plugin emulira

U ovom prvom primjeru tretirali smo jedan vrlo **glasan** trenutak na početku pjesme, tj. točno trenutak kada se dogodi prvi udarac po *snare* bubnju u pjesmi, i to smo provukli kroz FFT (Fast Fourier Transform), kako bismo dobili spektralnu karakteristiku identičnog „trenutka“ u pjesmi promatrano kroz prizmu frekvencije i glasnoće. Primjetna je razlika između ove tri valne karakteristike – prva (originalni zvuk bez efekata) je pri tome značajno drugačija od druge (CUDA plugin) i treće (hardverski uređaj). To je zapravo i očekivani rezultat, pošto je smisao CUDA plugin-a da bude što vjernija rekreacija načina kako hardverski uređaj radi. Stoga, nije nam toliko zanimljiva usporedba originalnog zvuka bez efekata koliko između druge i treće valne karakteristike koje predstavljaju usporedbu CUDA plugin-a i hardverskog uređaja.

Vidljivo je da su valne karakteristike vrlo, vrlo slične, sve do cca 1kHz, gdje postaju dosta različite. Primjetna je i razlika u glasnoći, i hardverski uređaj je pri istim postavkama za nekoliko dB glasniji od CUDA plugin-a. Iz ovog izoliranog slučaja možemo izvući dva zaključka – plugin i hardverski uređaj doista imaju sličnu valnu karakteristiku, ali je zbog vjerojatne pogreške kod modeliranja plugin-a odabrana druga referentna točka za „nultu glasnoću“. Pri tome treba uzeti u obzir i činjenicu da nisu svi hardverski uređaji isti, poglavito ne uređaj o kojem ovdje konkretno pričamo – radi se o uređaju koji radi kompresiju signala koji je proizveden prije dvadesetak godina, tako da je lako moguće i da uređaj koji smo koristili u studiju nije na isti način umjeren, ili da je baziran na novijoj verziji uređaja. Zapravo, kada se podigne nivo signala koji je obrađen kroz CUDA plug-in, tada spektralna karakteristika iznad 1kHz odgovara gotovo 100%.

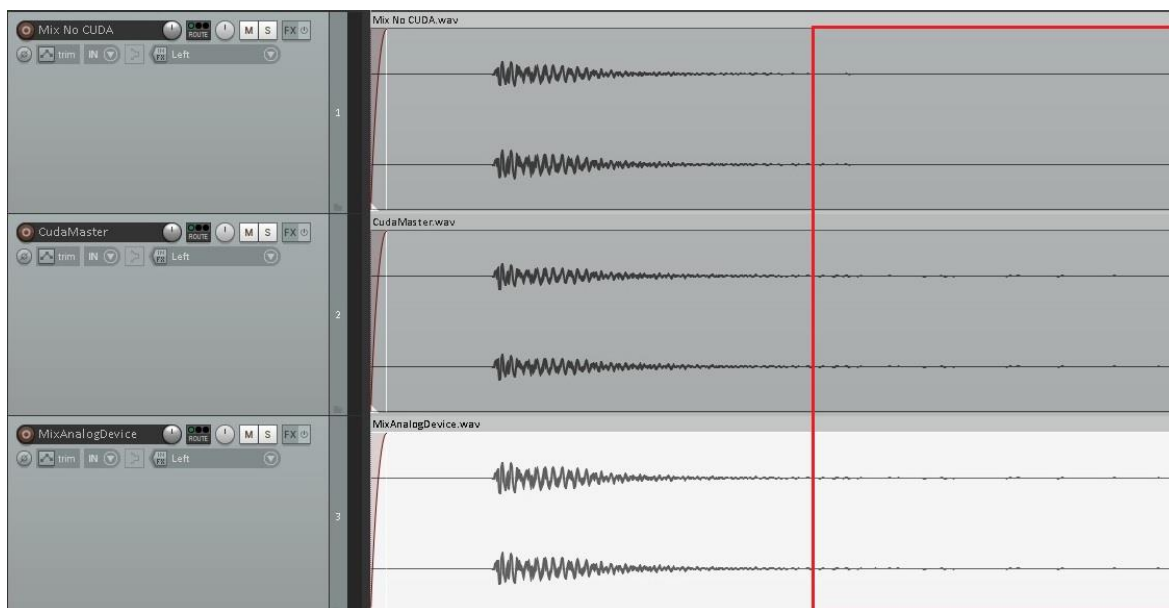
Slijedeći je primjer primjer direktne usporedbe između tri signala, ali u vremenskoj domeni, kroz prizmu glasnoće (bez promatranja frekvencije). Primjetna je identična karakteristika kao i u prethodnom testu – hardverski (analogni) uređaj ima jači signal (profesionalni glazbenici bi to protumačili kao jači *punch*, tj. većini bi se zbog veće glasnoće to više svidjelo). Čak dapače, ako usporedimo originalni signal, lako se primjeti da je korišteni analogni uređaj vrhunske kvalitete, pošto izuzetno malo zadire u signal, ali ga pojačava na tišim dijelovima, što je u skladu sa principom rada uređaja za kompresiju. Slično ponašanje potvrđeno je i pri korištenju EQ (ekvilizator frekvencija) plug-ina u usporedbi sa hardverskim EQ-om, što nam samo govori o tome koliko ovaj CUDA plug-in vjerno slijedi ponašanje hardverskog, analognog uređaja, koji je još k tome i izuzetno skup. Uzmimo u obzir i činjenicu da licence za ove softverske plug-ine koštaju svega nekoliko stotina dolara, a sam uređaj (kojeg zapravo ne možemo na jednostavni način koristiti više

puta u istoj pjesmi bez velikog gubljenja vremena), košta u rangu cijene 3.000EUR. Pogledajmo sliku:



Slika 4.2 Sva tri signala kroz prizmu glasnoće

Slijedeći je primjer isti onaj uvodni zvuk za koji smo analizirali spektralnu frekvenciju, ali opet kroz prizmu čiste amplitude. Naročito je zanimljiv način na koji CUDA plug-in i hardverski uređaj tretiraju nešto što se na originalnom signalu čini kao **tišina** (na slici označeno sa crvenom bojom). Naime, na originalnom zapisu se čini kao da je u označenom dijelu potpuna tišina, a korištenjem kompresora uspjeli smo iz dijela zvučnog zapisa gdje očitoma ima zvuka izvući dodatne artefakte i tranzijente koji su bili i u originalnom zvuku, ali su u procesu snimanja vjerojatno snimljeni nešto tiše pa ih se ne vidi na originalnom zapisu. Pogledajmo konkretan primjer:



Slika 4.3 Zvučni signal u amplitudnoj domeni, vrlo zanimljiv tretman "tišine"

Upravo radi ovakvih situacija i radimo kompresiranje zvučnog signala, pošto se zbog analogno-digitalne konverzije (ADC), različitih metoda kvantizacije zvuka i sl. može dogoditi da neki bitni tranzijent zvuka bude snimljen vrlo tiho. U takvim situacijama će kvalitetni kompresor zvuka efektivno „restaurirati“ neke tranzijente koji su bili vrlo tihi.

Vrlo slične efekte može se primjetiti i na CUDA plug-inu koji se koristi za ekvalizaciju zvuka (EQ) – vrlo je vjerno modeliran izvorni krug uređaja i samo na temelju vidljive razlike u amplitudi (glasnoći) moguće zaključiti da smo napravili ekvilizaciju signala. Pogledajmo primjer:



Slika 4.4 Amplituda i valni signal uz korištenje EQ-a

5. Zaključak

Impresivno je vidjeti koliko se softverski kod po kvaliteti emulacije približio hardverskim komponentama koje su desetljećima bile predmet opće pomame u studijima diljem svijeta. Gotovo da je u svijetu nemoguće pronaći vrhunski studio koji nema kompresor/EQ tipa Avalon VT-737SP, kojeg smo koristili kao bazu za usporedbu CUDA plug-ina i hardverske komponente.

S obzirom na postignute rezultate, bez zadrške se može preporučiti korištenje softverskih verzija već poznatih plug-ina, ali uz jednu važnu ogradu, zbog koje se moramo osvrnuti na tržište na kojem postoje stotine različitih emulacija jednog te istog hardverskog produkta.

Kada se pogleda realno, na tržištu softverskih plug-ina za obradu zvuka postoji zapravo samo jedno veliko ime i niz malih koji se pokušaju dokazati. Jedino veliko ime na tržištu plug-ina je Universal Audio, tvrtka sa desetljećima iskustva u razvoju hardverskih uređaja, čija je UAD platforma zapravo ASIC-verzija (Application Specific Integrated Circuit) onoga što je npr. Acoustica Audio napravila sa Nebula sustavom CUDA plug-ina. U međuvremenu se Acoustica Audio prebacila i na tržište CPU-baziranih plug-in-a, gdje ostvaruje čak i veće uspjehe od onih koji su već ostvarili kroz razvoj CUDA plug-in-a. Što samo pokazuje da na tržištu na kojem vlada softver stalna samo mijena jest, što je na kraju krajeva i dobro za korisnike.

Rezultati upućuju na činjenicu da se opisani softverski produkti mogu bez problema koristiti u vrhunskoj audio produkciji. To sa sobom nosi velike prednosti – manju cijenu, veću količinu instanci (za što bismo u slučaju korištenja hardverskih uređaja trebali veću količinu i po deset puta skupljih hardverskih uređaja), što donosi i veću fleksibilnost u radu. Također, hardverski uređaji trebaju i određenu dozu održavanja (promjena vakumskih lampi), čišćenja, teško se prenose (kvalitetni uređaji imaju veliku fizičku težinu). To je i jedan od razloga zašto su softverski plug-ini godinama toliko popularni. Na kraju krajeva, danas se za 50% cijene jednog opisanog uređaja može kupiti Universal Audio Apollo zvučna kartica sa osam ili više kanala, koja dolazi u paketu sa deset i više vrhunskih softverskih plug-inova, tj. emulacija hardverskih uređaja. Isto vrijedi i za proizvode cijelog niza drugih proizvođača, zbog čega je glazbena produkcija postala dostupnija za širi spektar korisnika, na čemu su svi korisnici sigurno zahvalni.

Popis kratica

CUDA	<i>Compute Unified Device Architecture</i>
DAW	<i>Digital Audio Workstation</i>
API	<i>Application Programming Interface</i>
GUI	<i>Graphical User Interface</i>
GPU	<i>Graphical Processing Unit</i>
CPU	<i>Central Processing Unit</i>
MKL	<i>Math Kernel Library</i>
IPP	<i>Integrated Performance Primitive</i>
STL	<i>Standard Template Library</i>
ALU	<i>Arithmetic-Logic Unit</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
CT	<i>Computed Tomography</i>
MRI	<i>Magnetic Resonance Imaging</i>
HOOMD	<i>Highly Optimized Objectoriented Many-particle Dynamics</i>
USDOE	<i>United States Department of Energy</i>
CFD	<i>Computational Fluid Dynamics</i>
DNN	<i>Deep Neural Network</i>
SDK	<i>Software Development Kit</i>
EQ	<i>Equalizer</i>

Popis slika

Slika 2.2.1 Paralelni rad CPU-a i GPU-a.	3
Slika 2.2 Programski model CUDA GPU-a	6
Slika 2.3 Memorijski model CUDA GPU-a.....	7
Slika 2.4 CUDA radni tok	8
Slika 3.3.1 Nebula3 Free	15
Slika 3.3.2 AcquaVox.....	15
Slika 3.3.3 Red Eq	16
Slika 3.3.4 Stradipad Free.....	16
Slika 3.5 Primjer sučelja softverskog DAW-a.....	18
Slika 3.6 Model arhitekture audio plug-ina	20

Literatura

- [1] GOUDARD, W.; MÜLLER, R. *Real-time audio plugin architectures*, Paris: IRCAM, 2003.
- [2] KEFAUVER, A. P.; PATSCHE, D. *Fundamentals of Digital Audio, New Edition*, Middleton, 2007.
- [3] GIBSON, D.; POLFERMAN, R. *An Architecture For Creating Hosting Plug- Ins Use In Digital Audio Workstations, International Computer Music Conference 2011*, Huddersfield, 2011.
- [4] NVIDIA, CUDA Zone, <https://developer.nvidia.com/cuda-zone>, pristupljeno 20. siječnja 2018.
- [5] SANDERS, J.; KANDROT, E. *CUDA by Example, An Introduction to General-purpose GPU Programming*, Boston, 2011.
- [6] NVIDIA, CUDA In Action – Research & Apps, <https://developer.nvidia.com/cuda-action-research-apps>, pristupljeno 20. siječnja 2018.

Prilog

Završni rad može imati priloge, ali se oni ne prilažu uz pisanu verziju završnog rada već se mogu priložiti na završnom ispitu ukoliko povjerenstvo na završnom ispitu tako odluči. Važno je čuvati svu poratnu dokumentaciju koja je nastala pri izradi završnog rada.

S unutarnje strane na zadnjim koricama originala, kao i svake kopije završnog rada, pričvršćuje se CD s kompletnim završnim radom u izvornom formatu (npr. .doc) i .pdf formatu sa svom popratnom dokumentacijom i programima. Pri čemu je obvezno da na tom CD- u postoji i dokument koji opisuje kako se rezultat njegova diplomskog rada (softver ili hardver) koristi (ili kako se npr. izvode mjerenja koja je opisao u radu). Ako se radi o softveru nužno je opisati i kako se programska podrška instalira.



Algebra

visoka škola za
primijenjeno računarstvo

KORIŠTENJE GRAFIČKIH KARTICA U OBRADI ZVUČNOG ZAPISA

Pristupnik: Kruno Karlović, 0065031583

Mentor: M. Sc. EE Vedran Dakić